Numerical Weather Prediction

Tests of the prpoposed semi-implicit integration scheme for the Unified model part 1: idealised Asian monsoon

Forecasting Research Technical Report No. 168

Mark Mawson

email: nwp_publications@metoffice.com

©Crown Copyright

Forecasting Research Technical Note No. 168

Tests of the proposed semi-Implicit integration scheme for the Unified Model Part 1: Idealised Asian Monsoon.

M. H. Mawson

June 1995

This document has not been published. Permission to quote from it must be obtained from the Director of Forecasting Research.

© Crown Copyright 1995.

Abstract

This note briefly describes the 3D version of the proposed semi-implicit integration scheme for the Unified Model and discusses various issues pertaining to its formulation. The model's current performance, with respect to both computationally efficiency and accuracy of simulation, is evaluated using an idealised Asian monsoon test problem. This work has lead to the current development version of the scheme, version 8, as described in Cullen, Davies & Mawson (1995).

1 Introduction

This is the second in a series of reports describing tests of the proposed semi-implicit integration scheme for the Unified Model, Cullen et al. (1995), in this case referring to the model as described in version 8 of that paper. In this report the dry formulation of the model is briefly presented and the following issues discussed;

- Formulation of the pressure gradient terms.
- Consistency of Helmholtz equation formulation and the application of the implied corrections.
- The elliptic solver.

Results from the first idealised 3D test problem are then presented. This involves simulating the Asian monsoon and East African Jet using a prescribed forcing function which is supposed to represent latent heat release over the Tibetan plateau, a simple representation of the East African highlands, and simple friction and Newtonian cooling. For the simulation with no mountains present a steady state solution exists, Sashegyi & Geisler (1987). When the mountain is present gravity waves excited by it prevent a steady state solution being obtained, although the solution changes little with time. This problem has been previously studied using the operational forecast model which ran on the CYBER 205 computer, Mawson & Cullen (1992).

2 The Dry 3D Algorithm

The dry 3D non-hydrostatic equations are

$$\frac{D\underline{u}}{Dt} = -f\underline{k} \times \underline{u} - C_p\theta\nabla\Pi - g\underline{k} + \underline{F} \tag{2.1}$$

$$\frac{D\theta}{Dt} = S \tag{2.2}$$

$$\frac{\partial\rho}{\partial t} + \frac{1}{\partial r/\partial\eta}\left[\frac{1}{cos\phi}\frac{\partial}{\partial\lambda}\left(\frac{\rho u}{r}\frac{\partial r}{\partial\eta}\right) + \frac{1}{cos\phi}\frac{\partial}{\partial\phi}\left(\frac{\rho v cos\phi}{r}\frac{\partial r}{\partial\eta}\right) + \frac{\partial}{\partial\eta}\left(\rho\dot{\eta}\frac{\partial r}{\partial\eta}\right)\right] = 0 \tag{2.3}$$

$$p = \Pi\theta\rho R \tag{2.4}$$

where D/Dt is the Lagrangian derivative, $\underline{u} = (u, v, w)$, the density ρ contains a factor of r^2, S and $\underline{F}$ are forcing functions, Π is the Exner pressure, and

$$\dot{\eta} = \frac{1}{\partial r/\partial\eta}\left[w - \frac{u}{rcos\phi}\frac{\partial r}{\partial\lambda} - \frac{v}{r}\frac{\partial r}{\partial\phi}\right] \tag{2.5}$$

$$\nabla\Pi = \left(\frac{1}{rcos\phi}\left.\frac{\partial\Pi}{\partial\lambda}\right|_r, \frac{1}{r}\left.\frac{\partial\Pi}{\partial\phi}\right|_r, \frac{1}{r}\frac{\partial\Pi}{\partial r}\right) \tag{2.6}$$

where $|_r$ denotes a horizontal derivative taken along a constant r surface. The form that this equation should take when the co-ordinate surfaces are not flat is discussed in section 3.1.

The solution algorithm is as follows:

Step 1: Use equations (2.1) and (2.2) to produce estimates of $\underline{u}$ and θ at time level $n+1$ using the time level n values only and the semi-Lagrangian advection scheme, (Cullen et al. 1995, Mawson 1995).

Step 2: Require that the equation of state holds at time level $n+1$. Define corrections $X' = X^{n+1} - X^n$ where X is any of the data variables, u, v, w, Π, θ or ρ, and write the equation of state in terms of the values at time level n and the corrections. The procedure is to then linearize the equation of state in terms of the corrections and to write all the corrections in terms of Π'. This results in a variable coefficient Helmholtz equation for Π' which is solved via an iterative method, see section 3.3. Some compromises have to be made in

forming the corrections and the result of this is that we obtain an estimate of Π' rather than the exact value. The correction to Π is then used to calculate $\underline{u}'$.

Step 3: Calculate p^{n+1} from $\Pi^n + \Pi'$ and ρ^{n+1} from the continuity equation (2.3). These terms agree exactly with those implied in the creation of the Helmholtz equation. In an incompressible case this ensures exact removal of the mass divergence.

Step 4: Calculate $\underline{u}^{n+1}$ and θ^{n+1} using equations (2.1) and (2.2) and the semi-Lagrangian advection scheme with the advecting variables and trajectory calculated using an approximation to $\underline{u}^{n+1/2}$ given by $\underline{u}^n + \alpha\underline{u}'$, where ideally the time-weight $\alpha = 0.5$. This step is not consistent with the way these terms are treated in the formation of the Helmholtz equation, see section 3.2 for a discussion of this, and as a result the equation of state is not exactly satisfied at time level $n + 1$.

Complete details along with the finite-difference forms of the equations can be found in Cullen et al. (1995). It may be beneficial to first read Coulter, Mawson & Cullen (1995) which describes the 1D vertical version of the scheme.
This procedure can be understood by considering a much simpler problem which explains in essence what the algorithm is doing. Consider simply

$$\frac{DQ}{Dt} = 0$$

Lagrangian advection of some quantity Q. Suppose we know where a particular blob of Q is at time t and we wish to know where it is at time $t + \Delta t$. To calculate this we need to know the trajectory it followed, ie we need to know what the velocity field is over the time interval $(t, t + \Delta t)$. Given this velocity field we can solve our equation. This is essentially what the new integration scheme is doing, except that it has to first calculate the velocity field over the time interval. The first two steps of the algorithm provide a good estimate to the velocity field at time $n + 1$ and this is used along with the value of the velocity field at time n to give the velocity field over the entire time interval. We then simply solve all the advection equations. The

first two steps only provide a good estimate, not an exact answer, because of the difficulty in solving step 2 exactly. In principle we could iterate the procedure to obtain better and better estimates of the velocity field at time-level $n+1$, but as the error in the estimate is probably not significant compared to other model errors it is unlikely any benefit would accrue from this.

3 Discussion of algorithm

3.1 Pressure gradient term.

The horizontal derivative parts of the pressure gradient term, equation (2.6), are defined on a constant r surface. The model uses a terrain following co-ordinate and hence it is necessary to define how we calculate the pressure gradient term on a constant r surface in the model. Since the λ and ϕ terms are similar we consider just the ϕ term. There are two alternative ways to calculate this term. The first is to simply interpolate Π to the desired r surface and then take finite-differences, the second is to expand the desired term via the chain rule,

$$\left.\frac{\partial \Pi}{\partial \phi}\right|_r = \frac{\partial \Pi}{\partial \phi} - \frac{\partial r}{\partial \phi}\frac{\partial \Pi}{\partial r}$$

where $|_r$ represents a derivative on a constant r surface and the horizontal derivatives on the right-hand-side are taken along co-ordinate surfaces. Either of these schemes could in principle be used in the model. Using second-order centred finite-differences in the chain rule version and linear interpolation in the interpolating version produces similar schemes, identical if Π is a linear function of r. The problem with the interpolating scheme comes when it is not possible to interpolate Π, ie: when the desired r surface is below the bottom model level, which occurs at level 1 whenever the surface is not flat. The desired Π can be estimated on this r surface by hydrostatic extrapolation. However there is a still a problem as we also need to define Π' on this surface. This cannot be estimated accurately by a hydrostatic extrapolation, as would be required to be consistent with the estimation of Π^{n+1} on the r surface by hydrostatic extrapolation. The alternative is to use some form of boundary condition, but this is also inconsistent with the hydrostatic extrapolation

of Π^{n+1}. Using the interpolating scheme in the Asian monsoon test problem with either hydrostatic extrapolation of Π' or Neumann boundary conditions produces model failures within the first day. The chain rule form is easier to implement, since all that is required as a definition of the centred difference of $\partial\Pi/\partial r$ across the bottom model level, and this can be provided by a one-sided difference. The same approach is easily applied to Π' and is automatically consistent with the definition of the derivative using Π^{n+1}. Runs of the Asian monsoon test problem execute satisfactorily with this form of the pressure gradient term. The chain rule form has been adopted as the pressure gradient term definition in Cullen et al. (1995).

3.2 Helmholtz equation.

The purpose of the Helmholtz equation is to calculate corrections to the velocity field used to provide the initial estimates of the data at time level $n+1$ so that the equation of state is satisfied at time level $n+1$ by the data calculated at time level $n+1$ using the initial velocity field plus these corrections. To do this exactly requires the solution of a highly non-linear equation and is not practical. We instead solve the equation linearized in the corrections, as this is much simpler, but accept that this will only provide an estimate of the solution and to obtain the exact solution will require the entire procedure to be iterated, as noted previously in section 2. Ideally we would like to form the linearized equation to be as consistent as possible with the advection scheme. However, since we are using a semi-Lagrangian advection scheme this is difficult to do as it requires the trajectories to become part of the Helmholtz equation. To avoid this complexity the Helmholtz equation is formed as an Eulerian correction to a semi-Lagrangian first guess. However the corrections we apply to the fields in the correction step are not evaluated using the same Eulerian correction but by a second semi-Lagrangian advection step. This introduces an inconsistency between these two steps and could lead to convergence problems, particularly near the ground where the Eulerian finite-differencing and semi-Lagrangian trajectory limiter can give quite different answers. This could easily be removed by using the Eulerian update rather than the semi-Lagrangian update and has the advantage of being cheaper, but has several disadvantages. To obtain similar temporal accuracy

it would be advisable to calculate the first guess using an estimate of the velocities at time level $n+1/2$ which would require storage of the velocities at time level $n-1$ in order to obtain an estimate at time level $n+1/2$ by extrapolation. It also requires a conservative update to θ to be used in the Helmholtz equation which would mean giving up the energetically consistent form of the Helmholtz equation. It would also mean giving up the monotonicity property of the semi-Lagrangian advection step as well.

A second inconsistency is introduced by the use of different definitions of θ' inside the Helmholtz equation. The different definitions are used in one case to obtain energetic consistency, and secondly for simplicity to reduce the cost and complexity of the Helmholtz equation, see Coulter et al. (1995) for the 1D vertical example of this. These two different definitions, besides being inconsistent with each other, are inconsistent with any way of performing the update to the potential temperature field that follows.

A third area of inconsistency is introduced by omitting certain terms when writing the corrections to the fields in terms of the correction to Π. The removal of these terms helps produce a more strongly elliptic Helmholtz equation, but again introduces an inconsistency between the formulation used to derive the Helmholtz equation and the update procedure which follows. The inclusion of more of these missing terms is primarily a matter of affordability versus benefit, since the extra ellipticity of the Helmholtz equation may not be significant. Cullen (1989) showed that omitting some of these terms was essential in getting a balanced model to find the correct solution in certain situations. These terms may also need to be omitted from the new integration scheme when using long timesteps where the solution should be similar to that obtained from a balanced model. It could be argued that the most important terms are the pressure gradient terms, coriolis terms and acceleration terms, and all these are currently included, and that the omission of some of the advective terms is of little importance. This argument is hard to prove conclusively, and it is hard to devise a numerical problem which would suitably demonstrate the importance of any of the missing terms.

It is hard to quantify the effect of the inconsistencies we have introduced in form-

ing the Helmholtz equation differently to how the updates to the fields are then performed. Since the inconsistencies are purely in terms of corrections, which in a slowly evolving flow will be very small $O(\Delta t^2)$ at most, it is possible that these inconsistencies and missing terms introduce errors which are insignificant and smaller than those introduced by linearizing the equation of state. The net effect of them is probably no more than a slight increase in the the error of the estimate of the corrections to be applied to the velocities. This error could be reduced by iterating the procedure to obtain successively better estimates. However as we have already stated it is unlikely that that any significant benefit will be gained from this due to other model errors being more significant. We would like to have no inconsistencies at all but to achieve this would require the abandonment of energetic consistency, a more complicated and hence more expensive to solve Helmholtz equation, the loss of monotonicity in the advection of potential temperature and wind fields since only a simple Eulerian scheme for the corrections can be included easily in the Helmholtz equation formulation, plus modifications to the first advection step which include extrapolating the wind fields forward in time. Most of these are undesirable features and may be worse than having the inconsistencies.

3.3 Elliptic Solver.

The choice of which iterative solver is between a *multigrid* approach and a *conjugate gradient* type approach. The multigrid approach requires two basic building blocks for it to succeed. Firstly, a cheap approximate solver for the elliptic equation which removes the high frequency errors but not necessarily the low ones. Secondly a way of transferring the problem consistently from the initial computational grid to a succession of coarser ones. Finding the first building block is made difficult by two things. Firstly our choice of a regular latitude/longitude grid on the sphere causes anisotropies to appear and forces us to use more expensive approximate solvers to remove the error. Secondly, the pressure gradient terms over the orography require some extra complexity to be added to the approximate solver. The second building block is relatively straight forward except over the orography. This is due to the changed shape of the boundary as the grid is coarsened. Similar problems are being

encountered in using multigrid in computational fluid dynamics for modelling flow past complicated geometries and solutions are beginning to be found. For the moment it seems sensible to monitor the progress with such work and return to the idea of using a multigrid approach at a later time. We are thus left with using a method from the conjugate gradient family or a similar variational approach. These methods can suffer from poor convergence properties unless suitably pre-conditioned. The method we have adopted for the moment is the Generalised Conjugate Residual method Eisenstat, Elman & Schultz (1983), GCR(k) for short. This method is described below along with some of its properties and failings. The search for a better method is being undertaken with help from the University of Oxford.

GCR(k) works by considering the error in the solution and searching along orthogonal directions removing the error in each direction as it goes. The maximum number of possible search directions is equal to the number of nodes in the problem and hence the number of iterations can also be equal to the number of nodes. The actual number of iterations required is equal to the number of distinct eigenvalues of the elliptic operator since this is the number of distinct search directions. This can still be a very large number. To reduce this number a suitable pre-conditioner could be used. Currently no pre-conditioner is applied. The search directions used by GCR(k) also need to be orthogonal to one another. This can only be guaranteed if the operator is symmetric. This can be achieved for the second derivative terms in the Helmholtz equation, but not for the first derivative terms, or vice versa. It is found that the more symmetric the operator the better the convergence properties. Currently the operator is re-scaled to ensure that the vertical second derivative term is symmetric. Assuming that we are given an elliptic equation

$$L(s) = R \tag{3.7}$$

where L is a negative definite elliptic operator on $\mathbb{R}^n$ and R and s belong to $\mathbb{R}^n$, then the GCR(k) algorithm is as shown in figure 1. Two slight modifications are made to the convergence testing. In the basic GCR(k) algorithm, as shown in figure 1, convergence is defined as occurring when the error becomes less than a specified value. This test this been modified so that convergence is defined to occur when

$$r < \epsilon \|L(s^0) - R\|$$

for some tolerance ϵ. This is a more sensible target for computational purposes since machine precision will limit the value of ϵ which is achievable. A second, optional, convergence criteria is also introduced. Rather than checking the residual we check the implied increment and define convergence if

$$\|\beta r^v\| < \epsilon_{inc}$$

for some tolerance ϵ_{inc}. In practice the algorithm does not find a monotonic decreasing set of increments and to avoid an erroneous definition of convergence due to a particularly small increment this criteria is applied to a specific number of consecutive increments, the number chosen being a user supplied parameter. This definition of convergence is potentially more useful since it is in terms of an increment to Π and hence to p and it is easier to decide what is a meaningful level of convergence to request.

4 Idealised Asian Monsoon.

The problem is that studied in Mawson & Cullen (1992). The potential temperature profile is based on that of a typical tropical atmosphere, Stevens, Lindzen & Shapiro (1977), with

$$\theta_{ref}(p) = \theta_0 + \frac{\Gamma}{\Pi\kappa} - \frac{\Gamma}{\kappa}$$

where $\theta_0 = 300K$ and $\Gamma = 620g/R$ is the mean static stability. The model is forced by a heating profile given by

$$H(\lambda, \phi, p) = B(p) e^{-\left(\frac{\lambda-\lambda_0}{\lambda_w}\right)^2} e^{-\left(\frac{\phi-\phi_0}{\phi_w}\right)^2}$$

where (λ_0, ϕ_0) defines the centre of the heating function, chosen to be at $24^o N$,$82.5^o E$, and λ_w, ϕ_w define half-widths for the heating function of 22.5^o and 12^o respectively. The vertical term is defined as

$$B(p) = \begin{cases} Q_c \frac{(p-p_t)(p-p_a)(p-p_b)}{(p_m-p_t)(p_m-p_a)(p_m-p_b)} & p_t \le p \le p_b \\ 0 & \text{otherwise} \end{cases}$$

where $p_t = 14000pa$ is the top of the heating function, $p_b = 90800pa$ is the base of the heating function, and $p_a = 93200pa$ is chosen so that the maximum heating

occurs at $p_m = 40000pa$. The maximum heat per day is given by $Q_c = 6K/day$. This heating function is an idealised representation of the latent heating due to convective cloud. The model also includes Newtonian cooling to represent radiative cooling back to the reference state given by

$$N(\lambda, \phi, p) = -N_c[\theta(\lambda, \phi, p) - \theta_{ref}(p)]$$

where $N_c = 2.314 \times 10^{-6} s^{-1}$, ie a 5 day time-scale. The total forcing to θ in equation (2.2) is given by

$$S(\lambda, \phi, p) = H(\lambda, \phi, p) + N(\lambda, \phi, p)$$

Simple friction is also included at every level in the horizontal so that

$$F = (-C_d u, -C_d v, 0)$$

with $C_d = 2.314 \times 10^{-6} s^{-1}$, ie a 5 day time-scale. The only orography is a simple representation of the East African Highlands given by a $2Km$ ridge defined between $15^o S$ and $15^o N$ and $27 - 34.5^o E$. The model is initialised to a basic state at rest and then run for 10 days using a timestep of 3600 seconds at a resolution of $96 \times 65 \times 17$ where the vertical levels are chosen to equally spaced in pressure with the bottom pressure level at $94444pa$ and the top θ level at $2777pa$.

This problem without orography has a steady state solution and the steady state is reached by about day 8 of the integration. The solution consists of a low pressure centre displaced a few degrees west of the maximum heating and was calculated by Sashegyi & Geisler (1987) for a linearized model and by Mawson & Cullen (1992) for primitive equation and semi-geostrophic models. The model as run here is of higher vertical resolution than that of Mawson & Cullen (1992) but similar horizontal resolution, and we do not expect any major differences in the solution obtained here except for possibly better vertical resolution of the low level jet along the orography. The primary use of this problem is to verify the new integration scheme works with orography and to test the effect of varying the time-weighting parameters. There are four time-weighting parameters, α_i, $i = 1, 2, 3, 4$, which affect the weighting of the update to u and v, the update to w, the pressure gradient terms in the u and v momentum equations, and the pressure gradient term in the w momentum equation

respectively. The ideal choice of the time-weighting parameters is $\alpha_i = 0.5$ for every value of i, which gives a second-order accurate in time scheme but this value is exactly on the linear stability limit and the scheme may not be stable. Three runs are described;

Run 1: $\alpha_i = 0.5$. Elliptic solver convergence criteria are, increment tolerance $= 1 \times 10^{-8}$, residual tolerance $= 1 \times 10^{-6}$ or 450 iterations reached.

Run 2: As run 1 but increment criteria not used.

Run 3: As run 1 but $\alpha_i = 1.0$.

The pressure field at 514 metres, the bottom θ level at day 8 is shown for the first two runs in figure 2. Both fields show signs of high latitude noise, although it is much worse in run 1. This noise is produced initially by the elliptic solver not converging to machine precision. It is well known that the convergence rate for solvers on equally spaced latitude/longitude grids on the sphere is worst in the neighbourhood of the poles and this noise is not totally unexpected. It is hard to say whether the growth of this noise in these runs is due to continually poor solutions produced by the elliptic solver in the neighbourhood of the poles or whether it is also partly due to having all the time-weights set to 0.5 and the model being susceptible to instability in high latitudes. Run 1 fails at 233 hours whilst run 2 successfully reached 10 days. Run 3 also runs successfully to day 10 and the pressure field at 514 metres for it and run 2 can be seen in figure 3. It is immediately obvious that run 3 has no noise in high latitudes despite using the less accurate form of the elliptic solver. This is because there is more implicit diffusion in the scheme with the time-weights set to 1 as it is only first order accurate in time. However the main pressure centre is very similar in both runs, as expected since this is the steady state part of the solution and should be unaffected by the choice of the time-weights. Further work will be done to try to improve the solver and establish whether the model is susceptible to instability in high-latitudes with the time-weights all set to 0.5. To look at the solution obtained we consider run 3, since it is noise free and the choice of time-weights has little or no impact on the underlying solution. The pressure field at 514m, figure 3 shows a low with central pressure approximately 8hpa lower than its initial value and with

very similar shape to that obtained with the primitive equation model in Mawson & Cullen (1992), henceforth OPE. The pressure field does not contain the erroneous large scale wave at the equator that was produced in OPE. Figure 4 shows the u field at 514m. The features to note are the strong flow around the northern end of the orography and the flow across the orography at the equator. The flow around the northern end of the orography is enhanced because of the blocking affect of the mountain and can also be seen in OPE. The flow across the orography at the equator is also a feature of the OPE solution. Figure 5 shows the v field at 514m. The obvious feature is the flow along the orography on the eastern side mimicking the East African Jet, again a feature of the OPE model. figure 7 shows a cross-section of v along the equator from $0 - 90^0$E. The strongest jet is at the surface as expected with a secondary maxima near the top of the mountain. A similar result was gained in OPE. The maximum values of v are very similar despite the slightly higher vertical resolution used in the new integration scheme run. This was not unexpected given the still very low horizontal resolution. The weak cross-equatorial return flow balancing the low level mass transport is visible at higher levels and is much smoother than that obtained in the OPE model. The u cross-section, figure 6, shows a strong flow across the mountain at the equator, which was also found in the OPE model.

4.1 Computational performance.

The cost in seconds per timestep for the three runs are shown in table 1. The costs are in seconds per timestep since run 1 only completed 233 timesteps rather than the 240 timesteps of the other two runs. The differences in cost represent the different amount of time taken by the elliptic solver as all the other costs are the same. The increased time is due trying to remove extra residuals introduced by the high latitude noise in runs 1 and 2. To obtain a better breakdown of the costs we show output for the first timestep of run 1 which uses the increment tolerance convergence criteria for the elliptic solver, table 2. The table is split into 5 blocks; the elliptic solver, interpolation and departure point calculation, the rest of the model except, diagnostics, and set up plus forcing code for this problem. The total

time for the timestep was 13.1 seconds, of which 7.167 seconds was spent in the solver and 4.736 seconds was spent in interpolation and departure point calculation. The second of these figures is not unexpected since this is the major area of cost in a semi-Lagrangian scheme. The first figure represents a cost per iteration of GCR(k) of 0.0338 seconds. Since the megaflop (Mflops in table) rate of the code is very good we can only improve performance by reducing the amount of work done. The most scope for this is in finding a faster, and in few of the results obtained in the previous section, more accurate solver. The major cost in the interpolation scheme is the routine interpolation itself. This routine calculates where on the grid the point to interpolate to is, ie: which points it lies between. This requires several **IF** tests and there is some scope for improving it. Currently the code is written in modular blocks and to keep this structure simple involves recalculating some of the departure points unnecessarily. This extra work will be removed at some point in the future to save approximately $15 - 20\%$ of the departure point cost. Cheaper interpolation schemes, such as the quasi-cubic scheme used at ECMWF, could also be implemented to further reduce costs. Further work will be done in investigating the choice and performance of departure point and interpolation schemes. The total cost for a timestep is significantly higher than the current Unified Model integration scheme which costs about 0.5 seconds per timestep, and significant reductions will be required. The main target areas will be the solver and the interpolation and departure point code. These areas will be looked at in detail in the future.

Run 1	17.25
Run 2	22.41
Run 3	13.80

Table 1:

5 Conclusions and recommendations.

- The chain rule form of the pressure gradient term is adopted in preference to the interpolating form. The interpolating form does not work due to problems

with extrapolation into orography and self-consistency.

- The current formulation of the Helmholtz equation introduces some inconsistencies with the update step which is performed after it. To remove the inconsistencies several undesirable options must be taken which may be more detrimental to the solution then the inconsistencies are. These inconsistencies are probably small when compared to the linearized equation of state and their net effect is to make the estimate of the velocity fields at time level $n+1$ slightly worse. This impact could be offset by iterating the procedure which calculates the estimate of the velocities at time level $n+1$, but this is unlikely to be worthwhile as the errors are probably much smaller than other model errors.

- The current choice of elliptic solver, generalized conjugate residual with no pre-conditioning has poor convergence properties, in particular in polar regions. A better solver will be sought in conjunction with the Department of Mathematics at Oxford University.

- The model successfully simulated an idealised Asian monsoon and low-level East African Jet using a 2km mountain ridge and simple heating function. The results are in agreement with those found previously and show a reduced level of gravity wave noise. There is a problem with high latitude noise when the time-weights are set to 0.5 which leads to model failures. It is not yet clear whether this is due to the poor performance of the solver in high latitudes or whether the model is unstable with this choice of the time-weights. This will be investigated further when a more accurate version of the solver is available. For the moment the problem will be overcome by using a time-weighting parameter of greater than 0.5.

- The model is currently very expensive. The primary costs are the elliptic solver and the interpolation and departure point calculations. Work will be done in both these areas to try to find cheaper but just as accurate or more accurate algorithms, since there is little or no scope for saving time by code optimisation.

References

Coulter, S., Mawson, M. H. & Cullen, M. J. P. 1995 , The 1d, vertical, version of the proposed semi-implicit integration scheme for the unified model: Formulation and test results, FR Scientific Paper No 25.

Cullen, M. J. P. 1989 , Implicit finite-difference method for modelling discontinuous atmospheric flows, *J. Comp. Phys.* **81**, 319–348.

Cullen, M. J. P., Davies, T. & Mawson, M. H. 1995 , A semi-implicit integration scheme for the unified model, FR Working Paper No 154. version 8.

Eisenstat, S. C., Elman, H. C. & Schultz, M. H. 1983 , Variational iterative methods for non-symmetric systems of linear equations, *SIAM J. Num. Anal.* **20**, 345–357.

Mawson, M. H. 1995 , The proposed semi-lagrangian advection scheme for the semi-implicit unified model integration scheme, FR Working Paper No 162. version 4.

Mawson, M. H. & Cullen, M. J. P. 1992 , An idealized simulation of the indian monsoon using primtive-equation and quasi-equilibrium models, *Q. J. R. Meteorol. Soc.* **118**, 153–164.

Sashegyi, K. D. & Geisler, J. E. 1987 , A linear model study of cross-equatorial flow forced by summer monsoon heat sources, *J. Atmos. Sci.* **44**, 1706–1722.

Stevens, D. E., Lindzen, R. S. & Shapiro, L. J. 1977 , A new model of tropical waves incorporating momentum mixing by cumulus convection, *Dyn. Atmos. Oceans.* **1**, 365–425.

Name	Times Called	Total Time	Avg Time	% of total	Mflops
GCR_K	1	5.30E-01	5.30E-01	4.1	337.9
GCR_ELLIPTIC_OPERATOR	212	4.91E+00	2.32E-02	37.6	457.3
GCR_COEFFICIENT	421	1.04E+00	2.47E-03	8.0	229.0
GCR_TWO_NORM	423	6.87E-01	1.62E-03	5.3	174.2
INTERPOLATION	44	3.00E+00	6.81E-02	22.9	375.1
ROBERT	7	6.35E-01	9.07E-02	4.9	346.2
CUBIC_LAGRANGE	7	5.84E-01	8.34E-02	4.5	486.5
TRI_LINEAR	38	4.02E-01	1.06E-02	3.1	477.0
BI_LINEAR_H	238	9.85E-02	4.14E-04	0.8	405.4
MONO_CONSERV	1	1.66E-02	1.66E-02	0.1	191.0
DEPARTURE_POINT	7	2.86E-05	4.09E-06	0.0	0.0
PE_HELMHOLTZ	1	1.03E-01	1.03E-01	0.8	413.2
NEW_INTEGRATION_SCHEME	1	8.68E-02	8.68E-02	0.7	169.8
EXNER_CALC	1	6.51E-02	6.51E-02	0.5	265.4
SL_VECTOR_V1	1	4.03E-02	4.03E-02	0.3	350.4
SL_VECTOR_U1	1	3.91E-02	3.91E-02	0.3	337.7
SL_VECTOR_W1	1	3.70E-02	3.70E-02	0.3	338.7
SL_VECTOR_V2	1	3.63E-02	3.63E-02	0.3	362.5
SL_VECTOR_U2	1	3.50E-02	3.50E-02	0.3	349.9
SL_FULL_WIND1	1	3.50E-02	3.50E-02	0.3	386.6
SL_FULL_WIND2	1	2.30E-02	2.30E-02	0.2	399.1
FLUX_RHO	1	1.43E-02	1.43E-02	0.1	409.9
ETADOT_CALC	2	1.30E-02	6.49E-03	0.1	419.5
POLAR_VECTOR_WIND	153	1.55E-03	1.02E-05	0.0	110.6
SL_SCALAR_W_PTS	2	8.69E-04	4.35E-04	0.0	244.1
DIAGNOSTICS	2	5.30E-01	2.652E-01	4.2	184.1
FORCE_MONSOON	1	7.67E-02	7.67E-02	0.6	288.6
MONSOON_SETUP	1	1.30E-02	1.30E-02	0.1	205.4
MASTER	1	7.66E-03	7.66E-03	0.1	90.4

Table 2:

For any initial guess s^0 set

$$r^0 = L(s^0) - R$$

and defining convergence as the error $\|r^n\| < \epsilon$ for some $\epsilon \geq 0$

then

Do while $n = 1, 2, \ldots,$ *until convergence or maximum user requested iterations reached*

Do while $v = 0, \ldots, k-1$ *and convergence not obtained*

$$\beta = -\frac{< r^v L(r^v) >}{< L(r^v) L(r^v) >}$$

$$s^{v+1} = s^v + \beta r^v$$

$$r^{v+1} = r^v + \beta L(r^v)$$

if $\|r^n\| < \epsilon$ *then exit*

$$q = r^{v+1}$$

Calculate $L(q)$

$$\forall l = 0, v \quad \alpha^l = -\frac{< L(q) L(r^l) >}{< L(r^l) L(r^l) >}$$

$$r^{v+1} = q + \sum_{l=0}^{v} \alpha^l r^l$$

$$L(r^{v+1}) = L(q) + \sum_{l=0}^{v} \alpha^l L(r^l)$$

end do while

set $s^0 = s^k$, $r^0 = r^k$, *and* $L(r^0) = L(r^k)$

end do while

Figure 1: GCR(k) Algorithm

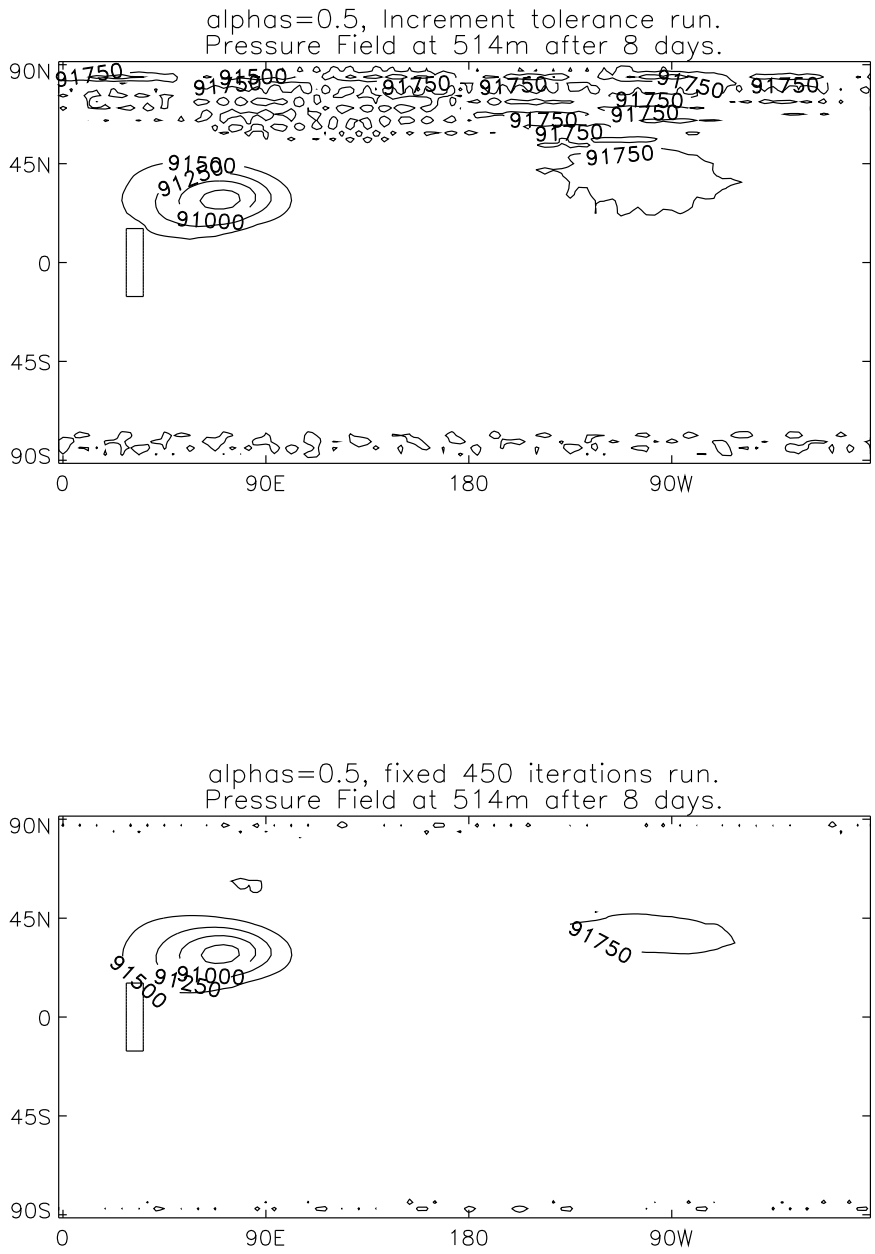
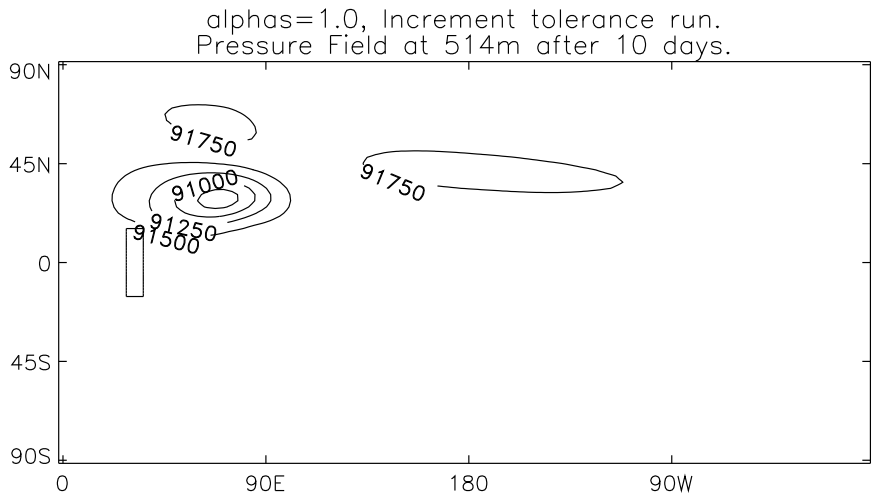


Figure 2: Contour interval $250pa$.



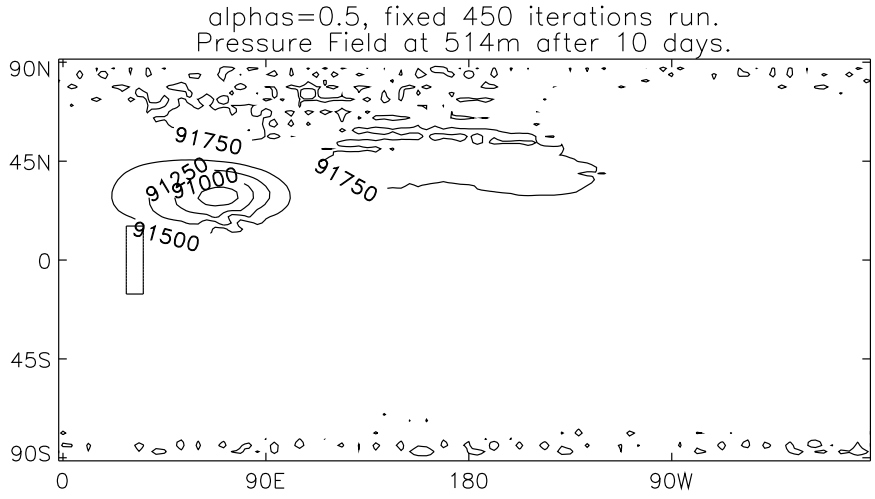


Figure 3: Contour interval $250pa$.

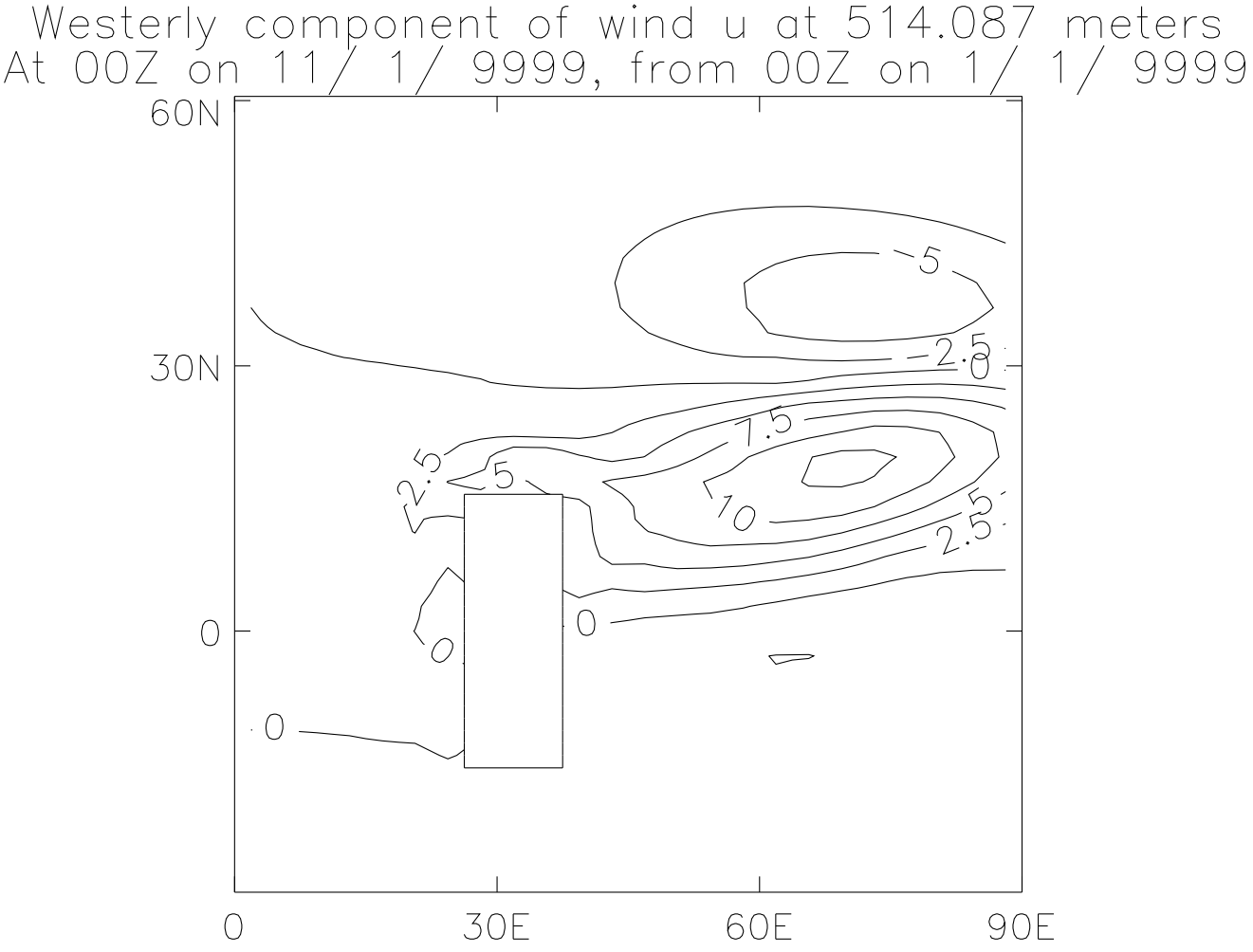


Figure 4: Contour interval $2.5ms^{-1}$.

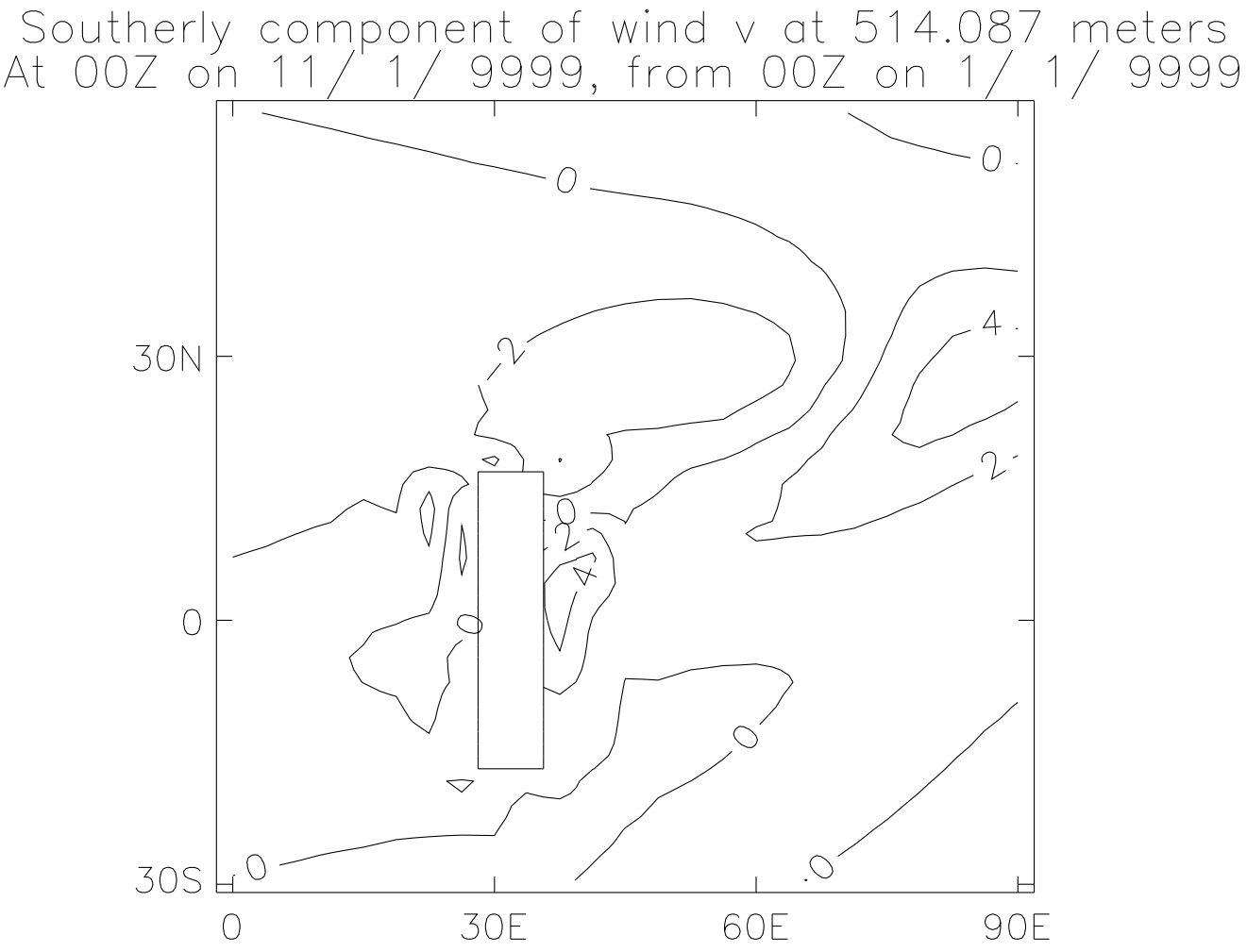


Figure 5: Contour interval $2ms^{-1}$.

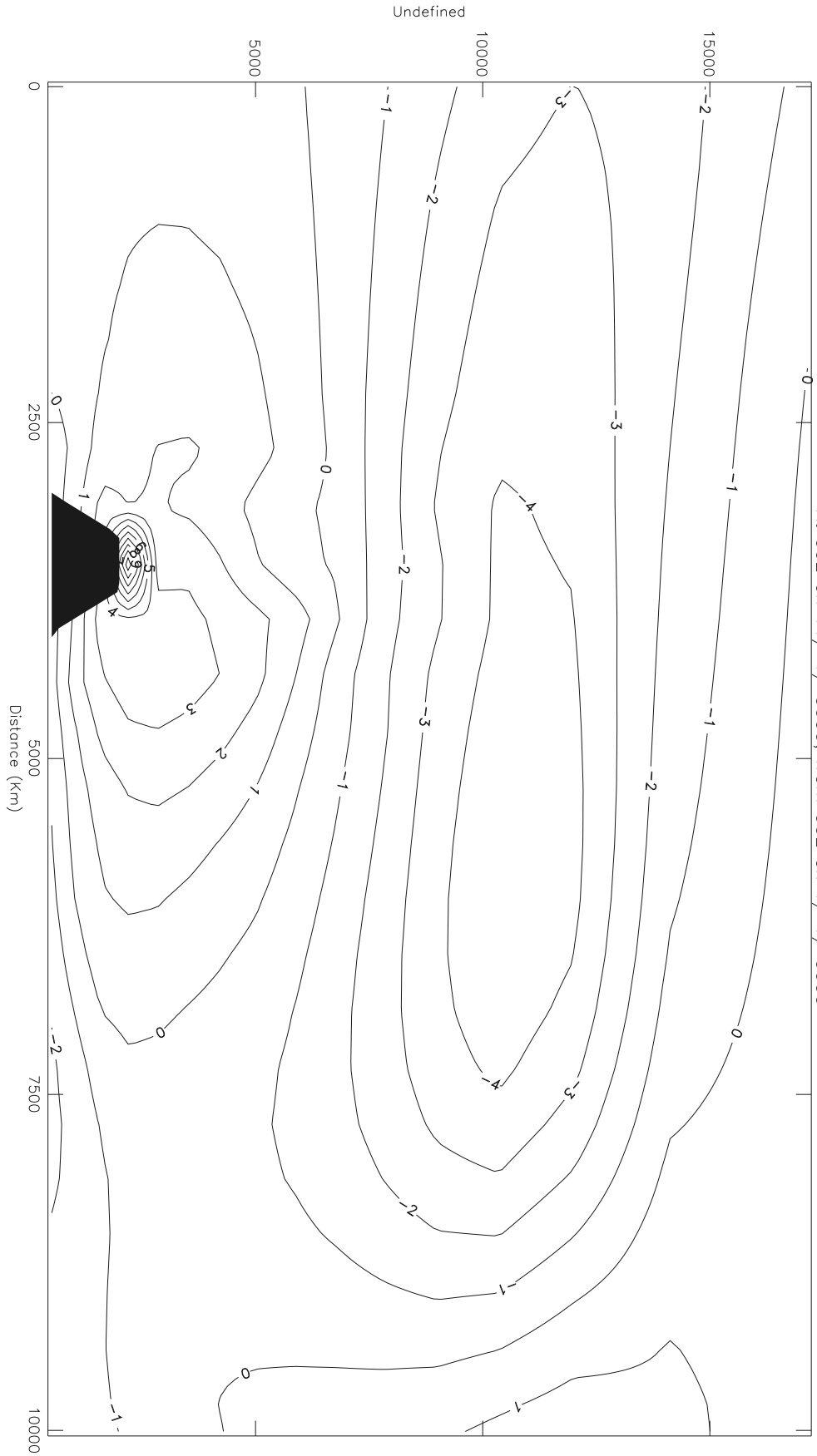


Figure 6: Contour interval $1ms^{-1}$.

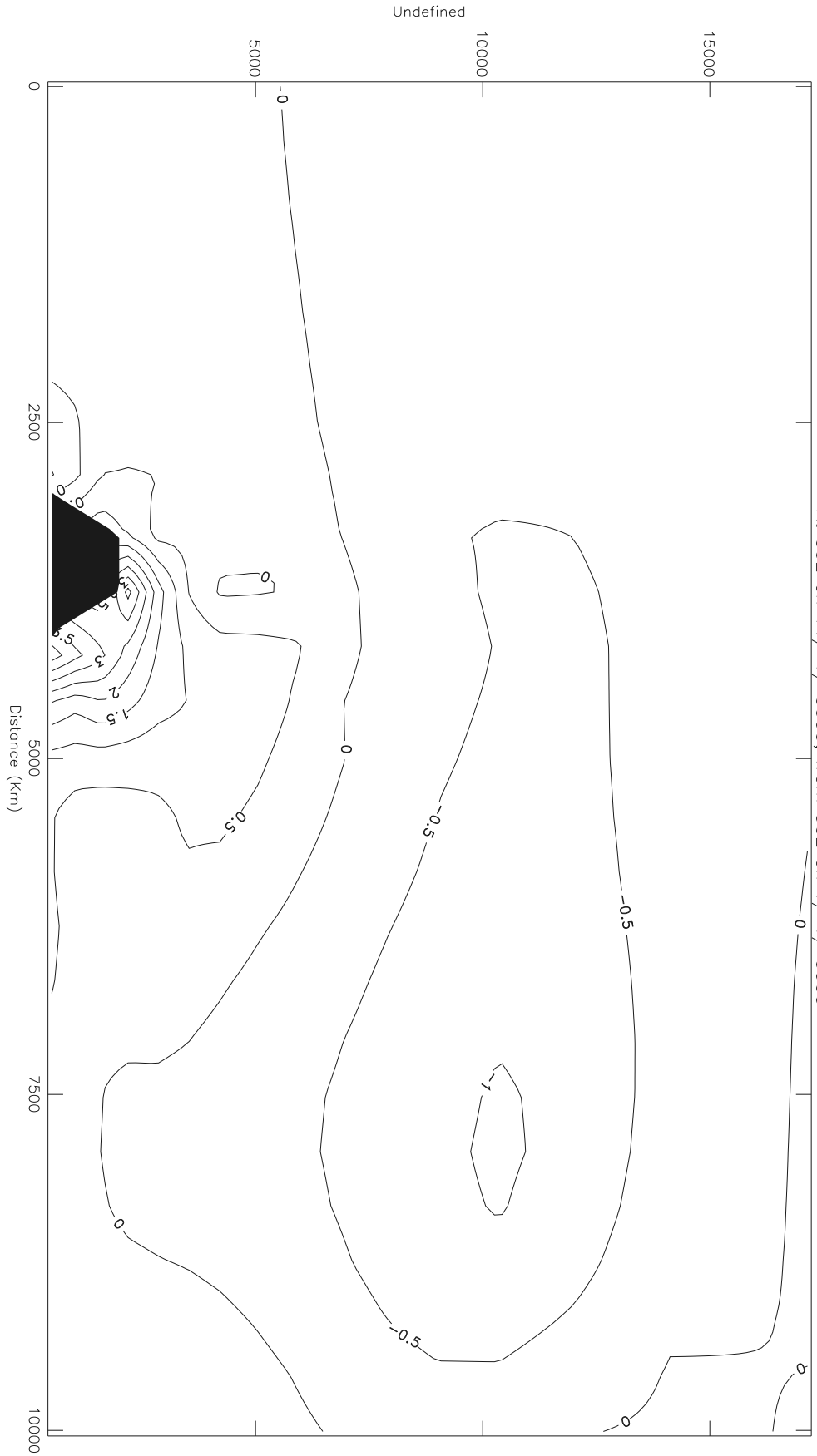


Figure 7: Contour interval $0.5ms^{-1}$.